

**TECHNICKÁ UNIVERZITA V LIBERCI**  
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2646 – Informační technologie  
Studijní obor: Informační technologie

## **Software pro řízení projektu**

## **Project management system**

### **Bakalářská práce**

Autor:	<b>Luboš Suk</b>
Vedoucí práce:	Ing. Roman Špánek, Ph.D.
Konzultant:	Ing. Pavel Tyl

V Liberci 17. 5. 2012



## Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum 17. 5. 2012

Podpis

## **Poděkování**

Rád bych poděkoval především vedoucímu své bakalářské práce, panu Ing. Romanu Špánkovi, Ph.D. za rady a trpělivost. Dále konzultantovi panu Ing. Pavlu Tylovi, který mi pomáhal upřesňovat nejasnosti v průběhu celé práce a panu Lukáši Seifertovi z firmy TRW Automotive za poskytnutí posudku k aplikaci.

Děkuji Vám.

## **Abstrakt**

Tato bakalářská práce se zabývá softwarem pro řízení projektu. V první části popisuje úvod do problematiky projektového řízení, zkoumá jednotlivé fáze a modely životních cyklů projektu, dále shrnuje postup při řízení softwarových projektů a programovou podporu projektového řízení.

Ve druhé části se zaměřuje na návrh samotné aplikace, která bude poskytovat minimální požadovanou funkcionalitu pro řízení softwarových projektů. Popisuje její jednotlivé části a průběh implementace. Rovněž obsahuje posudek aplikace který poskytla firma TRW automotive se sídlem v Jablonci nad Nisou.

## **Klíčová slova**

Software pro řízení projektu, projektové řízení, vývoj softwarových projektů, webová aplikace.

## **Abstract**

This thesis deals with project management system. In the first part it describes introduction to project management, researches different phases and project life-cycle models, the thesis further summarizes the procedure in the management of software projects and program support of project management.

In the second part it focuses on concrete suggestion of application, which will provide minimal needed functionality for project management. It describes its parts and process of implementation. It concerns also validation of application from TRW Automotive company based in Jablonec nad Nisou.

## **Keywords**

Project management system, project management, development of software projects, web application.

# Obsah

Prohlášení.....	3
Poděkování.....	4
Abstrakt.....	5
Klíčová slova.....	5
Abstract.....	6
Keywords.....	6
Obsah.....	7
Seznam obrázků.....	9
1. Úvod.....	10
2. Význam projektového řízení.....	11
3. Úvod do projektového řízení.....	14
3.1 Projekt.....	14
3.2 Projektový trojúhelník.....	15
3.3 Životní cyklus projektu.....	16
3.3.1 Koncepční fáze.....	16
3.3.2 Fáze návrhu.....	16
3.3.3 Fáze realizace.....	17
3.3.4 Fáze předání.....	17
3.4 Modely životního cyklu.....	17
3.4.1 Vodopádový model.....	18
3.4.2 Iterativní model.....	19
3.4.3 Inkrementální model.....	19
3.4.4 Spirálový model.....	20
3.5 Postup při řízení softwarových projektů.....	21
3.5.1 Analýza.....	21
3.5.2 Návrh.....	21
3.5.3 Implementace.....	21
3.5.4 Testování.....	22
3.5.5 Nasazení.....	22
4. Programová podpora projektového řízení.....	22
4.1 Nástroje nižší třídy.....	23
4.2 Nástroje střední třídy.....	24
4.3 Nástroje vyšší třídy.....	24
5. Návrh aplikace.....	24
5.1 Minimální funkcionalita.....	25
5.2 Návrh uživatelských oprávnění.....	26
5.3 Návrh databáze.....	27
6. Popis aplikace.....	28
6.1 Domů.....	28
6.2 Projekty a detail projektu.....	29
6.3 Úkoly a detail úkolu.....	30
6.4 Ganttův diagram.....	31
6.5 Přidávání projektů a úkolů.....	31
6.6 Komentáře.....	32
6.7 Soubory.....	33
6.8 Zprávy.....	33

6.9 Správa uživatelů.....	34
7. Implementace.....	35
7.1 Značkovací jazyk (X)HTML.....	35
7.2 Kaskádové styly CSS.....	36
7.3 PHP.....	37
7.4 Javascript.....	38
7.5 Databáze MySQL.....	39
7.6 Konkrétní způsoby realizace.....	39
7.6.1 Přihlašování.....	39
7.6.2 Úprava dat pro Ganttův diagram.....	40
7.6.3 Soubory na serveru.....	41
7.6.4 Ověřování vstupních polí formulářů.....	42
8. Validace.....	43
9. Závěr.....	44
10. Seznam literatury.....	45



## Seznam obrázků

Obrázek 1: Projektový trojúhelník.....	15
Obrázek 2: Vodopádový model.....	18
Obrázek 3: Iterativní model.....	19
Obrázek 4: Spirálový model.....	20
Obrázek 5: Use case diagram - uživatelé v aplikaci.....	26
Obrázek 6: Návrh databáze.....	27
Obrázek 7: Struktura menu aplikace.....	28
Obrázek 8: Domovská stránka z pohledu vedoucího.....	29
Obrázek 9: Detail projektu z pohledu vedoucího.....	30
Obrázek 10: Ganttův diagram.....	31
Obrázek 11: Zakládání projektu.....	32
Obrázek 12: Formulář pro vložení souboru.....	33
Obrázek 13: Výpis zpráv.....	34
Obrázek 14: Seznam uživatelů z pohledu administrátora.....	34
Obrázek 15: Tabulka bez stylování vzhledu.....	36
Obrázek 16: Tabulka nastýlovaná pomocí CSS.....	37
Obrázek 17: Ukázka ověřování přihlašovacích údajů.....	40
Obrázek 18: Identifikace hierarchie ganttova diagramu.....	41
Obrázek 19: Funkce pro mazání adresářů.....	42

# 1. Úvod

Předmětem zájmu této bakalářské práce je navrhnout a vytvořit pilotní implementaci aplikace, která bude splňovat minimální požadovanou funkcionalitu pro řízení softwarových projektů.

V první části této bakalářské práce je vysvětlen význam projektového řízení, a popsán stručný úvod do této problematiky, co se týče životních cyklů projektu, jejich modelů a postupu při řízení softwarových projektů. Dále je zde zmíněna programová podpora projektového řízení.

V druhé části je uveden návrh aplikace, který zahrnuje minimální požadovanou funkcionalitu, uživatelská oprávnění a návrh struktury databáze. Následně jsou popsány jednotlivé části vytvořené aplikace a jejich funkce. Poté se přistupuje k popisu samotné implementace zahrnující uvedení použitých technologií a uvedení několika příkladů konkrétní realizace. V závěru této části najdeme výsledné hodnocení aplikace ze strany firmy TRW Automotive.

## 2. Význam projektového řízení

V dnešní době si můžeme povšimnout nárůstu jak počtu řešených projektů, tak i jejich složitosti. I když pojem projektové řízení není nic nového, stále velké procento projektů končí neúspěchem. Softwarové projekty nejsou výjimkou.

[1] uvádí jako zajímavý příklad rozbor havárie záchranné služby v Londýně. Londýnský záchranný systém byl navržen pro automatické navádění vozidel záchranné služby. Každý řidič měl na palubě svého vozidla k dispozici počítačovou mapu a radiové spojení. V průběhu dnů 26. a 27. října 1992 se však systém zhroutil, což způsobilo dezorientaci řidičů, výjezd několika vozidel zároveň k jednomu případu, kolaps řídicího centra a přetížení tísňové linky. Tato situace měla za následek přímé ohrožení lidských životů. Možných příčin havárie je uváděno několik:

- **Nesmyslný návrh** – Projekt byl zadán dne 6.3.1991, výběrové řízení bylo ukončeno 15.4.1991, implementace systému proběhla do konce roku 1991, smlouva s dodavatelem byla podepsána v srpnu 1991, systém byl předán 8.1.1992.
- **Nezkušenost** – Celkem se o projekt ucházelo 17 realizátorů. Odborníci, kteří prováděli analýzu a zadání, doporučovali, aby byl požadován úplný systém za 1.5 milionů £ s dobou zpracování 18 měsíců. Byl však akceptován neúplný systém za 937 000 £, který byl dodán za 5 měsíců. Firma, která získala zakázku, použila CASE (Computer Aided Software Engineering) systém, který se teprve učila používat, navíc nikdy podobný systém nerealizovala a na tvorbu systému měla v době zadání pouhých 35 000 £.
- **Mnoho automatizace** – Lokalizace vozidel byla přímo podle hlášení případů, personál řídicího centra zasahoval až po 11 minutách. Neexistovala žádná písemná dokumentace. Předpokládala se stoprocentní spolupráce řidičů záchranných vozidel, přitom řidiči nebyli dostatečně ohodnoceni.
- **Uživatelské problémy** – Se samotnými posádkami vozů neprobíhaly při tvorbě systému žádné konzultace. Uživatelé byli vyškoleni dříve, než byl systém realizován. Systém neakceptoval prioritu operátorů v řídicím centru. Mnohem

častěji se stávalo, že lokalizace byla špatná, na rozdíl od předchozí hlasové komunikaci s operátory.

- **Neadekvátní komunikační systém a softwarové chyby** – Na kontrolních obrazovkách bylo mnoho informací, které však nebyly pro nedostatek paměti ukládány. V kritické dny nebyli ve službě žádní záložní operátoři. Systém byl 26.10. uveden do provozu se dvěma vážnými a 44 malými chybami. Programátoři zapomněli odstranit část ladicích tisků, která způsobila ztrátu informací na serverech. Celý systém byl napsán ve variantě programovacího jazyku Basic.
- **Špatný uživatelský interface** – Zprávy nepřetržitě rolovaly po obrazovce, aniž by je bylo možné zastavit, posádka vozu rovněž mohla velmi snadno na ovládacím panelu zmáčknout špatný knoflík, tiskárny bylo možné vypnout, aniž se zprávy někde ukládaly.
- **Špatný management projektu** – Nikdo nechtěl projekt vést, nikdo nepracoval na projektu na plný úvazek, programátoři nezvládli použitý CASE nástroj, softwarové změny se prováděly za pochodu, celý systém nebyl nikdy předem otestován.

Už jen samotný tento příklad ukazuje, že by se projektovému řízení měla věnovat daleko větší pozornost než dosud. I když se některé výše uvedené chyby mohou zdát nepochopitelné, praxe o tom hovoří jinak. Problémy se softwarovým řízením se projevovaly a stále projevují prodlužováním termínů, prodražováním projektů a výslednou nízkou kvalitou. [2] uvádí studii Standish Group Report, která byla zveřejněna v roce 1995 v USA, z níž by za zmínku stály následující informace (tabulky 1, 2 a 3 ).

Překročení nákladů o	Projektů
méně než 20 %	15,5 %
21 - 50 %	31,5 %
51 - 100 %	29,6 %
101 - 200 %	10,2 %
201 - 400 %	8,8 %
více než 400 %	4,4 %

*Tabulka 1: Překračování nákladů SW projektů*

Překročení času o	Projektů
méně než 20 %	13,9 %
21 - 50 %	18,3 %
51 - 100 %	20,0 %
101 - 200 %	35,5 %
201 - 400 %	11,2 %
více než 400 %	1,1 %

*Tabulka 2: Překračování času vývoje SW projektů*

Výsledná funkčnost	Projektů
méně než 25 %	4,6 %
25 - 49 %	27,2 %
50 - 74 %	21,8 %
75 - 99 %	39,1 %
100 %	7,3 %

*Tabulka 3: Dosažená funkčnost SW projektů*

Jak na těchto tabulkách vidíme, vyplývá z těchto informací, že průměrný softwarový projekt oproti původnímu plánu stál o 89 % více, vývoj trval 2,22 krát déle a výsledná funkcionality byla pouze 61 %.

### 3. Úvod do projektového řízení

Projektové řízení je oblast velmi obsáhlá, avšak důležitá pro správný průběh tvorby projektů. Projektový manažer, jakožto klíčová osoba pro úspěch projektu, by měl správně využít své znalosti projektového řízení k úspěšnému vedení projektu. Některé postupy projektového řízení je možné nastudovat z literatury, jiné je však možné získat pouze zkušenostmi z praxe v tomto odvětví. U rozsáhlých projektů samozřejmě přichází ke slovu softwarová podpora, která je u takovýchto projektů nezbytná.

V projektovém řízení se setkáváme s velkým množstvím pojmů a jejich definic. Úkolem této práce není všechny je vyjmenovat a popsat, ale pro lepší srozumitelnost jsou níže uvedeny základní poznatky k projektovému řízení.

#### 3.1 Projekt

Základním pojmem pro projektové řízení je *projekt*. Pod tímto pojmem si můžeme představit mnoho různých lidských činností. Může se jednat o velký projekt, na kterém se podílí mnoho osob, nebo i nejmenší formy projektů, na které stačí jedna osoba. V literatuře můžeme najít celou řadu definic, které se snaží tento pojem vysvětlovat. Definice z různých zdrojů se mohou lišit konkrétní formulací. Například [3] uvádí:

„Projekt je výsledek materiální nebo nemateriální povahy založený na strategickém plánu, navržený, organizovaný a realizovaný pod řízením někoho v zájmu vlastníka anebo zadavatele.“

„Projekt je aktivita omezená v čase, realizovaná pouze jedenkrát bez opakování se značným množstvím charakteristických rysů, ke kterým patří:

- výsledek musí sloužit užívání po celou dobu přesně určenou zadavatelem projektu,
- úspěch projektu při jeho startu není zřejmý,
- trvání projektu je časově omezeno,
- projekt je uskutečňován mimo běžnou podnikatelskou rutinu,
- zdroje pro realizaci projektu jsou limitovány,
- projekt má jen jeden výsledek.“

Zatímco ve [4] se dočteme:

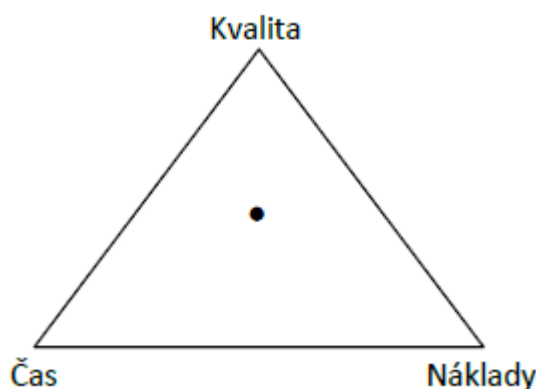
„Projekt je jakýkoliv jedinečný sled aktivit a úkolů, který má:

- dán specifický cíl, který má být jeho realizací splněn,
- definováno datum začátku a konce uskutečnění,
- stanoven rámec pro čerpání zdrojů potřebných pro jeho realizaci.“

„Projekt je dočasné úsilí vynaložené na vytvoření unikátního produktu, služby nebo určitého výsledku.“

### 3.2 Projektový trojúhelník

Při řízení projektů je nutno brát v úvahu čas ve srovnání s plánem, náklady ve srovnání se stanoveným rozpočtem a kvalitu projektu, která měří stupeň dosažených cílů [3]. Tyto tři základní ukazatele jsou navzájem propojeny viz obrázek 1 a nelze dosáhnout najednou maximální kvality projektu při minimálním čase a nákladech. Upřednostňováním jednoho z ukazatelů vždy zhoršuje kvalitu zbývajících dvou ukazatelů. Z toho vyplývá, že při důrazu na co největší kvalitu budou vysoké náklady a časová náročnost, protože v trojúhelníku by se tento projekt nacházel v blízkosti vrcholu Kvalita.



Obrázek 1: Projektový trojúhelník

### 3.3 Životní cyklus projektu

Projekt během svého vývoje prochází jednotlivými stádii, které se souhrnně nazývají životní cyklus projektu. Tento cyklus obvykle tvoří čtyři až osm základních fází, které mají specifické vlastnosti a vzájemně na sebe navazují. Počet fází se může lišit dle podrobnosti členění. [3] uvádí fáze čtyři, jakožto základní členění:

- koncepční,
- návrhu,
- realizace,
- předání.

#### 3.3.1 Koncepční fáze

V rámci této fáze identifikujeme potřeby a cíle, navrhujeme projektový tým, hodnotíme rizika, odhadujeme požadavky na zdroje a stanovujeme strategii vývoje projektu. Výsledkem této fáze by měla být studie proveditelnosti, která stanoví cíl, navrhne zásadní postup řešení a zhodnotí požadované zdroje pro dosažené cíle.

U softwarových projektu se jedná o zpracování požadavků a omezení od zadavatele, které je důležité formulovat co nejpřesněji. Výstupem bývá specifikace požadavků bez ohledu na druh implementace či výběr platformy. Špatná specifikace požadavků zpravidla mívá drtivý dopad na další úspěch projektu.

#### 3.3.2 Fáze návrhu

V této fázi dochází k vyhotovení detailního plánu projektu, jeho rozložení na jednotlivé činnosti s vyjádřením vzájemných vazeb, odhadem, jak dlouho bude trvat realizace činností a požadavků na jednotlivé zdroje. Zdroje jsou přidělovány k jednotlivým činnostem pro stanovení rozpočtu. Je nutno odhadnout rizikové faktory, které by mohly projekt jakýmkoli způsobem ohrozit. Projekt se vyjádří vhodným modelem (např. Ganttův diagram, síťový diagram) pro lepší představu o celkovém projektu.

V rámci softwarových projektů tato fáze navazuje na zpracování požadavků od zadavatele a rozšiřuje specifikaci o konkrétnější způsob implementace. Vybíráme zde například nejvhodnější strukturu systému, datový model, způsob provedení, uživatelské



rozhraní apod. Výsledkem této fáze by měl být projekt rozložený na jednotlivé úkoly a podúkoly spolu s hotovým časovým plánem.

### **3.3.3 Fáze realizace**

Při realizaci dochází k řízení a kontrole projektu. Řízení probíhá v reálném čase dle daného plánu a kontrolují se odchylky (čas, náklady, kvalita). Na základě těchto odchylek se zavádějí opravné opatření, aby se projekt příliš nerozcházel s plánem.

Softwarový produkt je v této fázi programován dle specifikace z předchozích etap. Tato fáze bývá zpravidla nejdelší a dochází k mnoha chybám. Proto je nutné důsledné testování jak jednotlivých částí produktu, tak i následně složených celků. Zjištěné chyby je nutné řešit co nejdříve nejen z finančních důvodů.

### **3.3.4 Fáze předání**

Předání je závěrečnou fází životního cyklu projektu, kde dochází k předání hotového produktu uživateli, poté je produkt spuštěn do provozu, testován a ověřen, zda správně plní účel. Celkový průběh je zpětně zhodnocen pro získání co nejvíce zkušeností k vývoji dalších projektů.

V této fázi by měl být softwarový produkt kompletní a připraven pro uvedení do provozu u zadavatele. Jelikož se prostředí pro běh produktu u zadavatele může lišit od prostředí, ve kterém byl produkt testován při vývoji, je nezbytné produkt po uvedení do provozu znovu otestovat. Na konci této fáze může být projekt ukončen, případně pokračovat na úrovni údržby produktu.

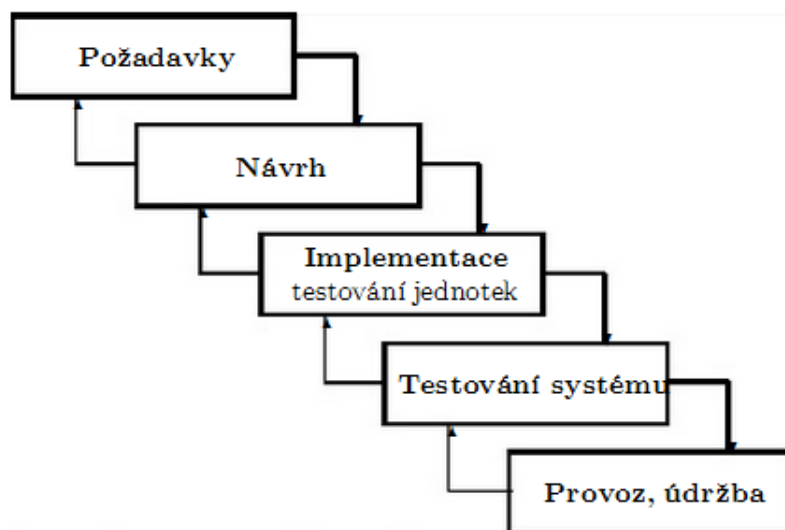
## **3.4 Modely životního cyklu**

Modely životních cyklů definují jednotlivé fáze projektu, jejich interakce, výstupy, popisují činnost průběhu systému a umožňují průběžnou kontrolu. Způsoby, jakými se má systém realizovat a jak budou jednotlivé fáze provedeny, je ponecháno na každé organizaci, protože konkrétní implementace se projekt od projektu vzhledem k jeho unikátnosti liší. [2] uvádí čtyři základní modely, které jsou posány v následujících podkapitolách.

### 3.4.1 Vodopádový model

Vodopádový model je základním modelem životního cyklu softwaru. Jednotlivé etapy jsou řazeny za sebou a každá následující etapa může být započata až po ukončení předchozí etapy. Jedná se o velice přirozený a zřejmě i nejstarší model životního cyklu softwaru.

V současné době už je tento model špatně použitelný z důvodu posloupnosti fází vývoje. Jako hlavní problém při použití tohoto modelu vystupuje zadavatel, který zpravidla není schopen předem stanovit všechny požadavky na software. Když pak máme projekt v závěrečné fázi, kdy je hotový a předveden, zadavatel vznesle celou řadu nových požadavků, případně se dožaduje změn původních. Tímto se u tohoto modelu dostáváme opět na začátek a musíme provádět jednotlivé kroky znovu, což je velice náročné jak časově tak i finančně.

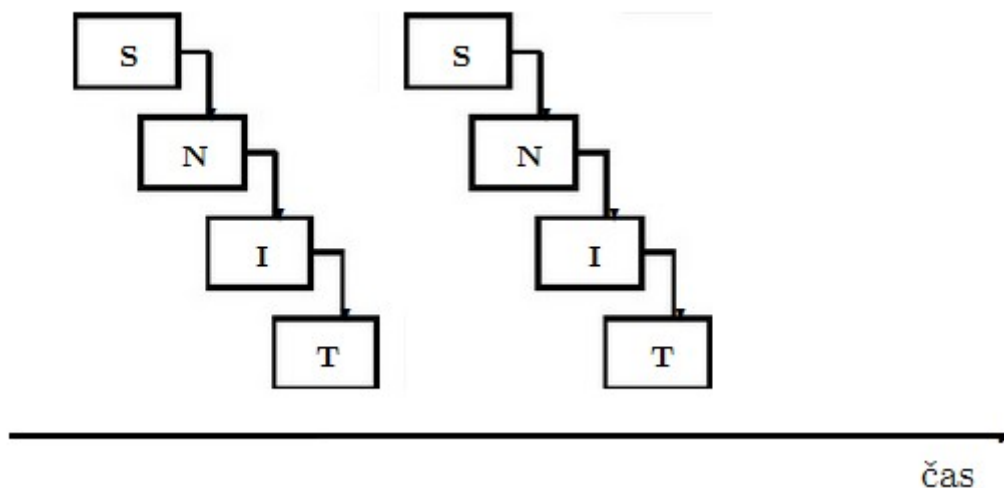


Obrázek 2: Vodopádový model

### 3.4.2 Iterativní model

Iterativní model se snaží odstranit hlavní problém vodopádového modelu rozdělením vývoje softwaru do tzv. iterací (viz obrázek 2., kde S = specifikace požadavků, N = návrh, I = implementace a T = testování).

Každou iteraci můžeme chápat jako jeden průběh vodopádového modelu. To znamená, že po každé iteraci má zadavatel k dispozici spustitelnou verzi softwaru (často s neúplnou funkcionalitou), kterou si může vyzkoušet a otestovat a ihned vznést připomínky či nové další požadavky. Díky tomu můžeme reagovat na změny od zadavatele dříve a lépe se jim přizpůsobit, to vše bez narušení životního cyklu softwaru.



Obrázek 3: Iterativní model

### 3.4.3 Inkrementální model

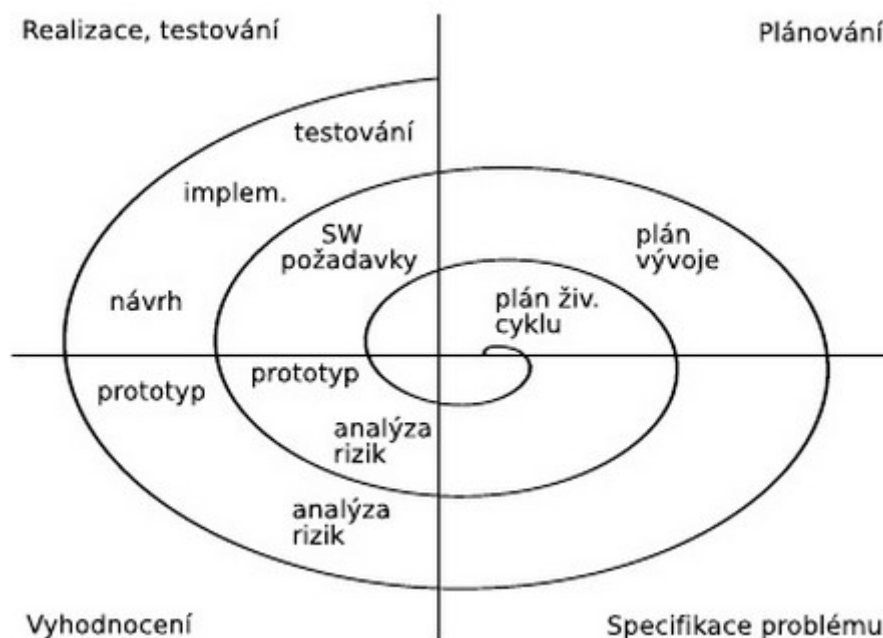
U tohoto modelu jsou ze specifikace stanoveny ucelené části systému, které se vytvářejí a odevzdávají zadavateli postupně. Tímto dáváme zadavateli možnost reagovat na jednotlivé hotové části. Případné další požadavky, nebo změny stávajících, nemají takový negativní dopad na vývoj softwaru jako u vodopádového modelu.

### 3.4.4 Spirálový model

Spirálový model do vývoje softwaru zavádí prototypování a klade značný důraz na analýzu rizik. Jednotlivé fáze životního cyklu se opakují pro každý prototyp, čímž připomíná iterativní nebo inkrementální model. Zde však zadavateli nepředkládáme k vyzkoušení verzi softwaru, ale pouze jeho *prototyp*. Po zakončení cyklu se posuzují dosažené výsledky oproti předchozímu cyklu.

Prototyp je po předvedení zadavateli zahozen a software (nebo další prototyp) je vytvářen znovu s tím, že máme k dispozici poznatky z tvorby předchozího prototypu. Tímto se s každým cyklem blížíme ke správnému výsledku za cenu vysoké spotřeby práce na tvorbě prototypů.

U spirálového modelu existují spustitelné verze softwaru již od začátku. To nám umožňuje mnohem jednodušší odhalování chyb a jejich následné odstraňování je daleko méně časově i finančně náročné. Spirálový model je však náročný na řízení a celkový přehled o projektu lze získat poměrně těžko.



Obrázek 4: Spirálový model

## **3.5 Postup při řízení softwarových projektů**

Řízení softwarových projektů je v podstatě koordinace zdrojů (lidských i materiálních) tak, aby bylo dosaženo stanoveného cíle projektu a zároveň nebyly překročeny náklady, termíny, kvalita a byly dodrženy stanovené požadavky. Probíhá v několika krocích, které jsou podrobněji popsány níže dle [5].

### **3.5.1 Analýza**

V průběhu tohoto prvního kroku se analyzují a upřesňují požadavky, které by vyvíjený software měl splňovat. Toho lze dosáhnout důkladnou spoluprací se zadavatelem, který pomocí jednání a sadou otázek umožní ujasnit zadání projektu, a na jejímž základě se poté vytvoří úvodní studie. Takto získaná studie je v podstatě předběžnou neformální specifikací požadavků, díky kterým je již celkem dobře možné udělat si představu, jakou funkčnost by systém měl poskytovat. Po sestavení úvodní studie se provádí rozplánování a rozpočet projektu, analyzují se možná rizika a stupeň proveditelnosti a následně se z těchto všech hledisek vyvozuje také celková nabídka.

### **3.5.2 Návrh**

Dalším krokem je návrh, který přímo vychází z předchozí výše popsané analýzy. Během návrhu se definuje, jakým způsobem by měl být software vytvořen. V tomto kroku jsou hlavním předmětem zájmu především dynamický a statický model softwaru, vytvoření uživatelského prostředí a také tvorba databáze. V průběhu návrhu se někdy objevují problémy, které jsou způsobeny zejména změnou požadavků, případně jejich nedostatečně detailní či dokonce špatnou specifikací. Tyto potíže vedou ku příkladu k nutnosti opakovaně upravovat datový model a provádět další opravy či změny, což je ovšem značně časově náročné, neefektivní a problematické pro celé projektové řízení.

### **3.5.3 Implementace**

Po úspěšném překonání a uzavření návrhu se dostáváme ke kroku, který nese název implementace. Tento krok se věnuje přímo samotnému vývoji softwaru, to znamená, že se přechází již k jeho programování. Implementace se řídí vždy podle detailního plánu, který vzniká až po ukončení návrhu, jelikož teprve v ten moment je

pevně určeno, jaké funkčnosti budou vytvářeny. Jedním z nejdůležitějších bodů je sestavení pořadí přírůstků, což probíhá pomocí hodnocení vývojového rizika a priority zadavatele. Ty přírůstky, které mají vysoké riziko a malou prioritu, se realizují první; přírůstky s vysokou prioritou i rizikem vyžadují naopak více času a pozornosti. Jakmile získáme pořadí přírůstků, lze sestavit již zmiňovaný detailní plán, který by měl určit strategii a způsob implementace, rozvrh programování přírůstků a jejich testování, integraci a rovněž integrační testování.

### **3.5.4 Testování**

Ve čtvrtém kroku přichází na řadu testování. Cílem tohoto kroku je ověření správnosti kódu softwaru a také testování, jestli chování softwaru odpovídá zadaným požadavkům. Během předchozího kroku implementace se totiž mohou v softwaru udělat chyby v důsledku nepřesností v komunikaci mezi vývojáři a podobným situacím. Právě testování by takové chyby mělo napomoci odhalit a odstranit.

### **3.5.5 Nasazení**

Posledním krokem vývoje softwaru je nasazení. Během tohoto kroku je software předán zákazníkovi, nainstalován a spuštěn do plného provozu. Do tohoto kroku někdy spadá rovněž dokončení kompletní uživatelské dokumentace a zaškolení samotného zadavatele, případně jeho pracovníků v užívání softwaru.

Následuje již jen ukončení a zhodnocení celého projektu, což je ovšem víceméně administrativního rázu. V úvahu může připadat také další vývoj, údržba nebo dokonce dlouhodobý rozvoj softwaru. V tom případě by se začal vytvářet další nový projekt.

## **4. Programová podpora projektového řízení**

V současnosti se v téměř všech oborech a skoro při všech úkonech či procesech používají počítačové programy. Představa, že by pro vývoj IT projektů neexistoval žádný software, který by projektové řízení usnadňoval, je tedy poměrně těžká. Otázkou ovšem je, co přesně a co vše by takový software podporující projektové řízení měl obsahovat a jak konkrétně by mohl práci ulehčovat. Například Milton D. Rosenau [6] přistupuje k softwarům pro řízení projektů takto: „Používat počítačový software k řízení

projektů není totéž jako efektivně řídit projekt. Nicméně s pomocí tohoto softwaru může dobrý manažer projektu odvést lepší práci. Naopak ze špatného manažera projektu se nestane dobrý manažer, když po něm budete chtít, aby používal software k řízení projektů”. Kvalitní software může uživateli nabídnout takové výhody jako včasné upozornění na možné riziko, naplánování a sledování průběhu celého projektu nebo kontrolu aktuálního stavu a porovnání s plánem projektu. Na druhou stranu software sám o sobě není schopen vyřešit různé potíže a problémy, které se v průběhu projektu mohou objevit. Z tohoto důvodu je řízení projektů zejména o řízení lidí.

Pokud je zvolen nevhodný nástroj pro řízení projektů, může způsobit víc škody než užitku. Častou chybou je snaha najít jakýkoli program, který bude schopen fungovat na současném vybavení. Naopak by se mělo nejdříve určit, jaký problém je potřeba vyřešit, a podle toho hledat vhodný software bez ohledu na vybavení, jakým tým disponuje. Potřebný hardware je možné vždy pořídit. Dále by program rozhodně neměl zbytečně zabírat čas a také je nutné počítat s dobou, kterou zabere zaškolení týmu, aby dokázali program správně a efektivně používat a rozuměli jeho výstupům.

Na poptávku po softwarové podpoře odpovídají projektoví manažeři a vývojáři softwaru. Stále ještě se najde mnoho lidí, kteří k projektovému řízení používají běžné aplikace a veškerou přípravu projektu (jeho rozsah, plán, rozpočet, přiřazování zdrojů, připravování dokumentace apod.) řeší pomocí tabulkového procesoru nebo textového editoru. Přitom existuje celá řada nástrojů a programů s mnoha specifickými funkcemi právě pro řízení projektů. V následujících podkapitolách jsou uvedeny základní typy dostupných nástrojů, které jsou rozdělené dle funkčnosti a ceny [7].

## **4.1 Nástroje nižší třídy**

Tyto nástroje nabízejí spíše jen základní funkce, při čemž se jejich cena pohybuje do 200 dolarů na uživatele. Jsou využitelné pro malé projekty nebo jednotlivé uživatele. Jejich největší výhodou je schopnost vykreslovat Ganttovy diagramy, které se bez vhodných nástrojů tvoří velmi špatně.

Mezi programy z této třídy můžeme uvést například Milestones Simplicity od společnosti KIDASA Software, Inc. Umožňuje jednoduché vykreslování Ganttových diagramů, najdeme zde také nástroj pro vytvoření osnovy projektu atd.

Jelikož se často používají běžné kancelářské produkty, nabízejí některé společnosti doplňky pro Microsoft Excel nebo Access, díky čemuž lze použít alespoň základní funkce pro řízení projektů.

## **4.2 Nástroje střední třídy**

Nástroje této třídy jsou schopné zvládat větší projekty a práci několika uživatelů na více projektech zároveň. Jejich cena se pohybuje v rozmezí 200-500 dolarů na uživatele. Nabízejí o poznání širší funkce jako tvorbu nejen Ganttových, ale i síťových diagramů, analýzy kritické cesty, sledování pokroku prací na projektech, možnosti tvoření zpráv o momentálním stavu a různé další.

Mezi nejrozšířenější nástroje střední třídy patří Microsoft Project, dalšími výrobci jsou společnosti Artemis, Primavera nebo Welcom.

## **4.3 Nástroje vyšší třídy**

Pro tento druh nástrojů se někdy používá označení software pro řízení podnikových projektů. Bývají integrovány s datábázovým softwarem a jsou dostupné rovněž přes Internet. Jejich součástí jsou rozsáhlé funkce použitelné pro velké projekty, spojují informace u jednotlivých projektů, díky kterým je možno vytvářet jakési souhrnné přehledy o průběhu daného projektu atd.

Jako zástupce této třídy můžeme jmenovat třeba Microsoft Enterprise Project Management Solution nebo z levnějších webově orientovaný VMPi Enterprise Online.

## **5. Návrh aplikace**

Cílem této části práce je navrhnout aplikaci pro řízení softwarových projektů a zachovat jednoduchost, uživatelskou přívětivost a pokrýt minimální požadovanou funkcionalitu.

Výsledná aplikace bude zpracována formou webové aplikace, aby k ní bylo možné přistupovat z libovolného počítače s internetem a internetovým prohlížečem bez vážného omezení funkčnosti. Bude využívat klasické třívrstvé architektury, kdy na webovém serveru je aplikační logika, na datábázovém serveru uložena konkrétní databáze s daty a uživatel se bude připojovat pomocí webového prohlížeče.



Aplikace bude vytvářena jako víceuživatelská s vhodně navrženými oprávněními. Uživatel by měl mít přístup jen k těm funkcím, které skutečně potřebuje a měl by mít odepřeny funkce, které nejsou pro jeho podíl na projektových pracích potřebné nebo by je dokonce mohl nevhodným počínáním ohrozit.

Funkce aplikace by měly pokrýt základní potřeby pro řízení softwarových projektů a zároveň co nejméně omezovat uživatele zbytečnostmi, které by aplikaci činily nepřehlednou.

## **5.1 Minimální funkcionalita**

Aplikace by měla poskytovat přehled všech projektů, které se v rámci organizace vytvářejí nebo vytvářely. Seznam by měl obsahovat základní informace o jednotlivých projektech jako například název, jméno vedoucího, termíny začátku a konce.

U jednotlivých projektů by měl být k dispozici detail, kde by byly zobrazeny veškeré informace týkající se projektu. K projektům by mělo být možné přidávat komentáře, soubory a jednotlivé úkoly, které je potřeba v rámci projektu splnit. Jednotlivé části projektu by měly být zobrazitelné pomocí Ganttova diagramu pro lepší přehled o časové náročnosti a návaznostech.

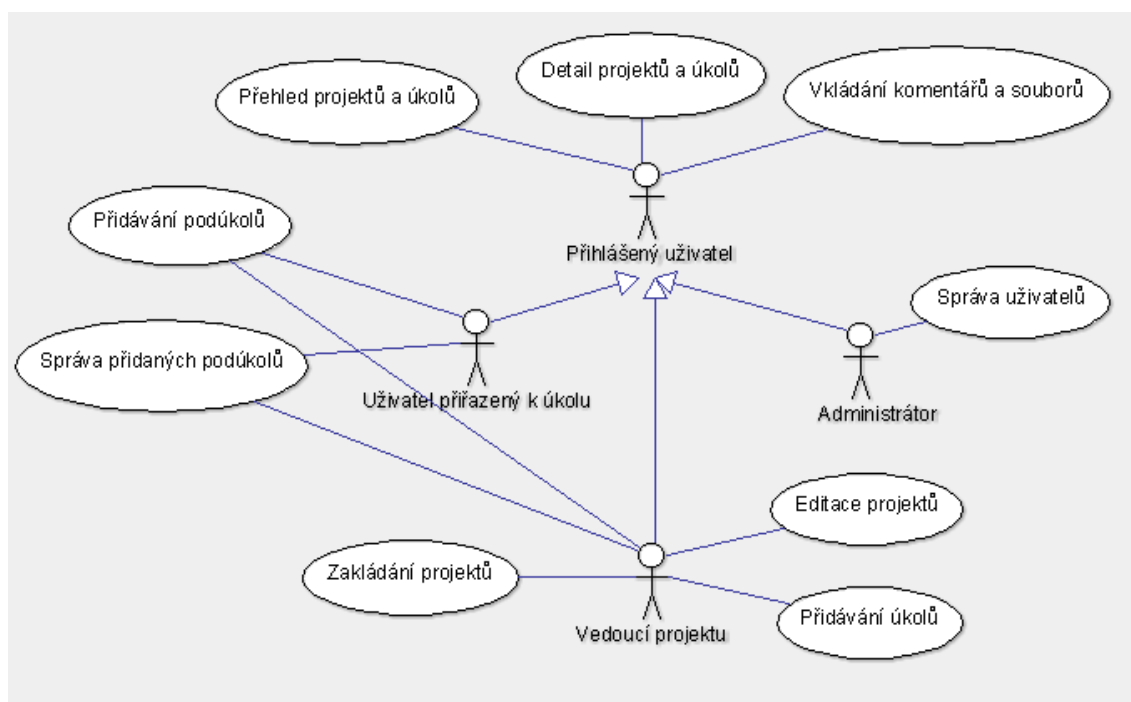
Dále by aplikace měla obsahovat veškerý přehled úkolů, který bude možno podobně jako u projektů třídit, a bude obsahovat základní informace o jednotlivých úkolech. Jednotlivé úkoly by mělo být možné dělit na podúkoly. Vzhledem k potřebě pružného řízení projektů by měl mít možnost přidávat podúkoly mimo vedoucího i uživatel přiřazený k řešení daného úkolu. Pro úkoly by měla být naimplementována možnost komentářů a správa souborů.

Uživatelé by měli mít možnost komunikovat v rámci aplikace. Například vhodnou implementací zasílání zpráv, která by jim umožnila lepší spolupráci bez nutnosti vázání komunikace na projekt. Zprávy, na rozdíl od komentářů, by nebyly veřejně přístupné pro ostatní uživatele.

## 5.2 Návrh uživatelských oprávnění

V aplikaci budou rozlišovány čtyři druhy uživatelů. Běžný uživatel, vedoucí a administrátor jsou nastavovány administrátorem. Čtvrtý druh uživatele bude určen přiřazením uživatele k úkolu vedoucím projektu. Uživatel může být zároveň vedoucím, administrátorem i přiřazen k libovolným úkolům.

Na obrázku 5 vidíme use case diagram znázorňující možné druhy uživatelů a jejich možností činnosti v aplikaci. Přihlášený uživatel má k dispozici přehled a detail projektů i úkolů a možnost vkládat soubory a komentáře. Uživatel přiřazený k úkolu má možnost k němu přidávat podúkoly, u kterých má pak umožněnou správu. Vedoucí má možnost přidávat nové projekty, spravovat ty, které vede a zároveň má veškerá práva nad úkoly, které náleží k jeho projektům. Administrátor má možnost přidávat nové uživatele a editovat údaje stávajících.

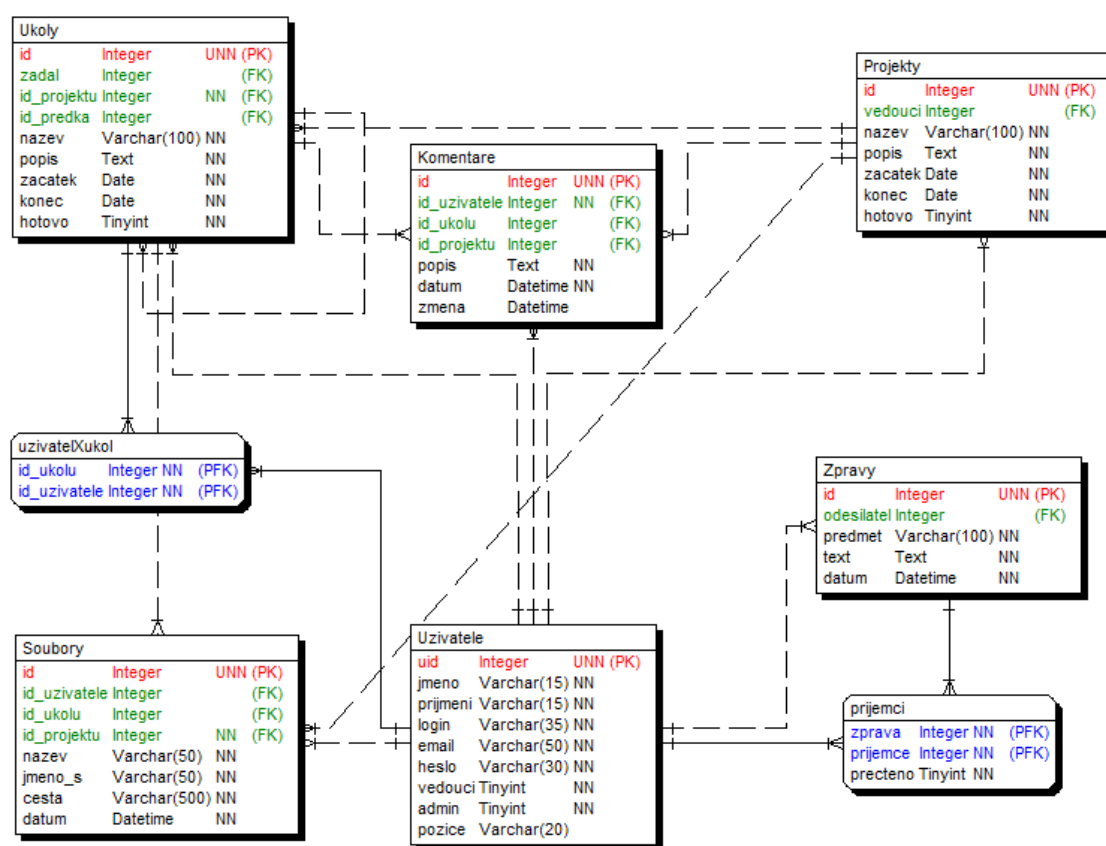


Obrázek 5: Use case diagram - uživatelé v aplikaci

## 5.3 Návrh databáze

Návrh databáze aplikace vidíme na obrázku 6 vygenerovaného pomocí programu Case Studio. Návrh je zobrazen ve formě, která je označována jako E-R diagram (Entity-relationship diagram).

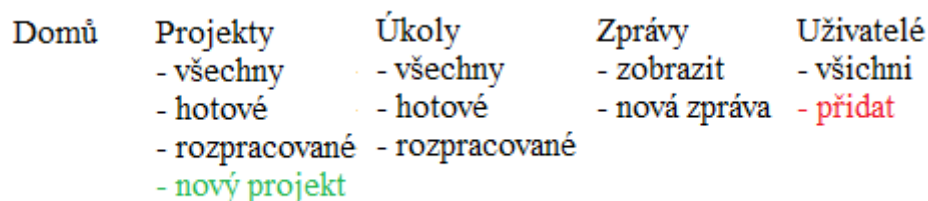
Atributy jednotlivých databázových tabulek jsou barevně rozlišovány. Červeně jsou uvedeny primární klíče, zeleně cizí klíče a modrou barvou jsou zvýrazněny primární cizí klíče relačních tabulek. Atributy, jejichž hodnoty nesmí nabývat nulové hodnoty, jsou označeny zkratkou NN (Not null). Označení U (unique) určuje, že hodnoty atributu v dané tabulce nesmí být duplicitní.



Obrázek 6: Návrh databáze

## 6. Popis aplikace

Tato část je věnována vytvořené aplikaci a jejímu popisu. Jsou zde rozebrány jednotlivé části. Na obrázku 7 vidíme strukturu hlavního menu aplikace. Černé položky jsou přístupné všem uživatelům, zelené jsou pouze pro uživatele, kteří jsou označeni jako *Vedoucí* a červené pro uživatele s označením *Administrátor*.



Domů	Projekty	Úkoly	Zprávy	Uživatelé
	- všechny	- všechny	- zobrazit	- všichni
	- hotové	- hotové	- nová zpráva	- přidat
	- rozpracované	- rozpracované		
	- nový projekt			

Obrázek 7: Struktura menu aplikace

### 6.1 Domů

Na domovskou stránku je uživatel přesměrován po přihlášení. Jejím úkolem je poskytovat uživateli základní přehled o úkolech, které mu jsou přiděleny. V případě uživatele s oprávněním *vedoucí* jsou zde zobrazovány veškeré projekty, které momentálně vede. V obou případech, jsou vypisovány pouze položky, které nebyly označeny jako *hotové*. Záznamy jsou zobrazovány v tabulce jak je vidět na obrázku 8. Výpisy v tabulkách je možné abecedně třídit.

U vypsaných projektů jsou ve sloupcích zobrazeny informace o názvu, vedoucím, začátku a konci projektu, počet splněných/celkových úkolů, stav projektu a zaškrťovací políčko pro smazání projektu se všemi jeho částmi.

Tabulka s úkoly obsahuje sloupce s informacemi o názvech úkolů, nadřazeném projektu, začátku a konci úkolu, stavu úkolu a zaškrťovací políčko pro smazání úkolu, pokud je to uživateli dovoleno.

Domů	Projekty	Úkoly	Zprávy	Uživatelé	Odhlásit	
------	----------	-------	--------	-----------	----------	--

**Projekty**

Název	Vedoucí	Začátek	Konec	Úkoly	Hotovo	
Projekt 01	Karel Kozelka	8.03.2012	23.03.2012	2/15	Ne	<input type="checkbox"/>
test barvy	Karel Kozelka	22.03.2012	22.03.2012	0/0	Ne	<input type="checkbox"/>
A zase je tu neděle	Karel Kozelka	22.03.2012	22.03.2012	0/0	Ne	<input type="checkbox"/>
Jak na krtky	Karel Kozelka	25.03.2012	25.03.2012	0/0	Ne	<input type="checkbox"/>
Test podcasti	Karel Kozelka	5.05.2012	5.05.2012	0/1	Ne	<input type="checkbox"/>

Smazat vybrané projekty

**Úkoly**

Název úkolu	Projekt	Začátek	Konec	Úkoly	Hotovo	
Strouhani mrkve	Projekt 01	22.03.2012	22.03.2012	0/0	Ne	<input type="checkbox"/>

Smazat vybrané úkoly

Obrázek 8: Domovská stránka z pohledu vedoucího

## 6.2 Projekty a detail projektu

Stránka projektů slouží pro přehled všech projektů, které byly v rámci aplikace vytvořeny. Zobrazení je provedeno stejným tabulkovým výpisem jako u projektů z domovské stránky viz obrázek 8. U výpisu je možné základní filtrování na všechny, hotové a rozpracované projekty.

Detailní zobrazení projektu nabízí veškeré informace týkající se daného projektu. Pod hlavním menu je umístěno projektové menu, které umožňuje zobrazit Ganttův diagram, přidávat komentáře a soubory pro všechny uživatele, a pro vedoucího daného projektu jsou zde navíc položky pro přidání úkolu a editaci projektu.

Základní informace, uvedení názvu projektu, jména vedoucího odkazující na jeho profil, termín začátku a konce projektu, popis a tlačítko pro smazání projektu, jsou umístěny přímo pod projektovým menu.

Domů
Projekty
Úkoly
Zprávy
Uživatelé
Odhlásit

Přidat úkol
Upravit projekt
Zobrazit gantt diagram
Přidat komentář
Přidat soubor

Název projektu: Ukázkový projekt  
Vedoucí: [Karel Kozelka](#)  
Od: 5.05.2012 do: 16.05.2012  
Popis:


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur neque nisi, venenatis fringilla molestie eu, iaculis in mauris. Phasellus semper nisi sed lectus tempus non ultrices libero porttitor. Suspendisse in massa diam. Suspendisse auctor massa vitae massa ornare luctus ultricies neque molestie.

Smazat projekt

Úkoly						
Název úkolu	Projekt	Začátek	Konec	Úkoly	Hotovo	
Nový	<a href="#">Ukázkový projekt</a>	5.05.2012	5.05.2012	0/0	Ne	<input type="checkbox"/>
druhá vrstva	<a href="#">Ukázkový projekt</a>	5.05.2012	5.05.2012	0/0	Ne	<input type="checkbox"/>

Smazat vybrané úkoly

Soubory


[soubor\\_01](#)  
Vložil: Karel Kozelka 7.05.2012 21:11

Komentáře

[Karel Kozelka](#) dne 7.05.2012 21:10  
[Editovat](#) [Smazat](#)  
Komentář...

Obrázek 9: Detail projektu z pohledu vedoucího

Pod základními informacemi je stránka rozdělena do tří částí, které jsou od sebe odděleny barevnou lištou s popisem. První je zde tabulka s přehledem úkolů, pod ní jsou vloženy soubory a na konci stránky komentáře k danému projektu.

### 6.3 Úkoly a detail úkolu

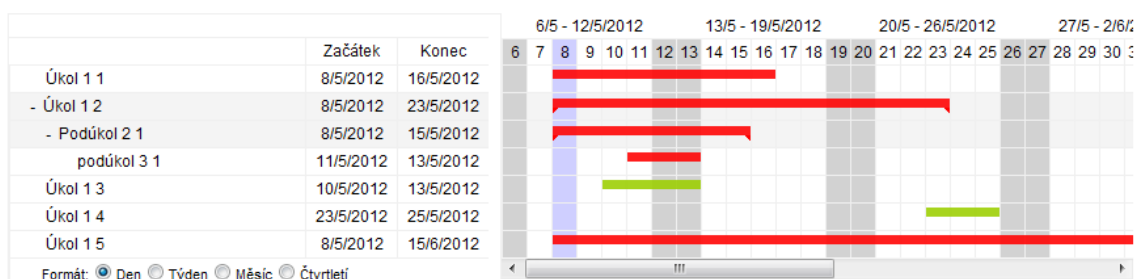
Stránka úkolů zobrazuje veškeré úkoly, které byly v rámci aplikace vytvořeny. Seznam úkolů je zobrazen v tabulce stejným způsobem jako u úkolů na domovské stránce viz obrázek 8. Zobrazení úkolů je možné filtrovat na všechny, rozpracované a hotové.

Detailní zobrazení úkolu je zpracováno stejným způsobem jako u detailu projektu viz obrázek 9. Navíc jsou zde uvedeny informace o zadavateli, nadřazeném projektu a seznam uživatelů, kterým byl úkol přidělen.

## 6.4 Ganttův diagram

Ganttův diagram lze zobrazit pro konkrétní projekt položkou z projektového menu. Zobrazuje veškeré úkoly a podúkoly přiřazené k danému projektu. Pro vykreslování Ganttova diagramu je použita modifikace nástroje jsgantt.

Na obrázku 10 vidíme, že v levé části diagramu jsou zobrazeny názvy úkolů a podúkolů s uvedenými daty začátku a konce. Položky v tomto seznamu fungují jako odkazy na úkoly. V pravé části diagramu jsou zobrazeny časové průběhy. Barevně rozlišujeme úkoly na červené nesplněné a zelené splněné. Pokud se jedná o normální linii bez špičatého ohraničení, pak úkol nemá definovány žádné podúkoly. Linie se špičatým ohraničením má k sobě definované podúkoly, které je možno zobrazit rozkliknutím znaménka + v levé části u názvu nadřazeného úkolu. Také je zde možnost formátu zobrazení průběhu úkolů na dny, týdny, měsíce a čtvrtletí.



Obrázek 10: Ganttův diagram

## 6.5 Přidávání projektů a úkolů

Pro zakládání nových projektů a úkolů slouží formulář. Na obrázku 11 vidíme formulář pro nový projekt. Tento formulář obsahuje pole pro název projektu, datum začátku a konce projektu a popis. Při zadávání úkolu je zde ještě seznam uživatelů, kterým může být úkol přidělen. Datum lze zadat ručně ve formátu DD.MM.YYYY nebo vybrat z vyskakovacího javascriptového formuláře kalendáře. Pro vkládání popisu slouží textové pole. S povoleným javascriptem je textové pole rozšířeno wysiwyg editorem Xinha, který uživatelům umožňuje pokročilejší práci s textem. Při odesílání formuláře je kontrolováno vyplnění názvu projektu a také správný formát data.

Název projektu

Začátek  Konec

Popis

-- size --

Cesta:

Zavřít kalendář

<<
>>

Květen 2012

Ne	Po	Út	St	Čt	Pá	So
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Obrázek 11: Zakládání projektu

## 6.6 Komentáře

Komentáře může přidávat každý přihlášený uživatel, a to ke všem projektům i úkolům. Vypisovány jsou pod sebe dle data vložení. U komentáře je vždy uvedeno jméno a příjmení autora odkazující na jeho uživatelský profil, datum vložení, tlačítka pro editaci nebo smazání, samotný text a informace o případné pozdější editaci komentáře.

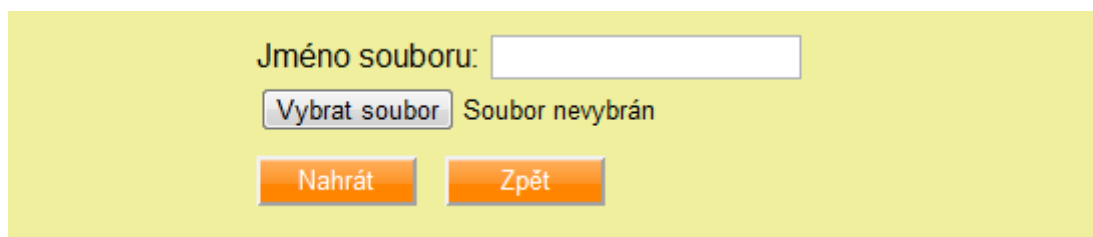
Psaní komentáře je přístupné z projektového nebo úkolového menu. Pro text komentáře je možné použít wysiwyg editor Xinha, umožňující lepší formátování textu.



## 6.7 Soubory

Soubory může vkládat každý přihlášený uživatel k libovolnému projektu nebo úkolu. U vloženého souboru je uveden název tohoto souboru, autor, datum vložení a tlačítko pro možnost smazání.

Vkládání souborů je přístupné z projektového nebo úkolového menu. Formulář pro výběr souboru obsahuje dvě položky. První položkou je jméno souboru, kde má uživatel možnost zadat, pod jakým názvem bude vložený soubor označován v detailu projektu nebo úkolu. Tato položka je nepovinná a bez vyplnění bude souboru zobrazován výchozím jménem. Druhá položka zobrazí uživateli průzkumníka pro přímé vybrání souboru, který bude stisknutím tlačítka Nahrát uložen na server.



Jméno souboru:

Soubor nevybrán

Obrázek 12: Formulář pro vložení souboru

## 6.8 Zprávy

Výpis všech přijatých zpráv je zobrazen v tabulce, kterou vidíme na obrázku 13. Výpis je realizován sortovatelnou tabulkou, která obsahuje sloupce s předmětem zprávy, jménem a příjmením odesilatele, data odeslání a checkboxy pro mazání zpráv. Nepřečtené zprávy jsou pro přehlednost barevně označovány.

Při psaní nové zprávy je uživateli zobrazen formulář s položkou pro předmět, výběrem uživatelů, kterým má být zpráva doručena a textovým polem pro samotný text zprávy.

## Zprávy

Předmět	Odesílatel	Datum	
Bez předmětu	Karel Kozelka	24.04.2012 14:57	<input type="checkbox"/>
Bez předmětu	Karel Kozelka	31.03.2012 19:43	<input type="checkbox"/>
Bez předmětu	Karel Kozelka	31.03.2012 19:43	<input type="checkbox"/>
sdfsdfs	Karel Kozelka	31.03.2012 19:43	<input type="checkbox"/>
Bez předmětu	Karel Kozelka	31.03.2012 18:57	<input type="checkbox"/>

Smazat vybrané zprávy

Obrázek 13: Výpis zpráv

## 6.9 Správa uživatelů

Na obrázku 14 vidíme tabulku se seznamem všech registrovaných uživatelů v aplikaci. Poskytuje základní informace jako jsou jméno, příjmení, email, pozici, oprávnění vedoucího a oprávnění administrátora.

Do aplikace není umožněna volná registrace a nové účty může přidávat pouze uživatel s oprávněním administrátor. Dále má možnost editovat a mazat stávající uživatelské účty.

### Uživatelé

Jméno	Příjmení	Email	Pozice	Vedoucí	Administrátor
Karel	Kozelka	karel@karlovic.cz	Nacelnik	Ano	Ano
Jarka	Nametelka	jarka@nametelka.cz	Neuvedeno	Ano	Ne
Choze	Nevim	choze@nevim.cz	Neuvedeno	Ne	Ne
Jaja	Paja	email@email.cz	Neuvedeno	Ne	Ne
Pepa	Netrefil	netrefil@vedle.cz	Neuvedeno	Ne	Ne
franta	vomacka	nevim@nevim.ne	Neuvedeno	Ano	Ano

Smazat vybrané

Obrázek 14: Seznam uživatelů z pohledu administrátora

## 7. Implementace

V následujících podkapitolách je uveden popis technologií použitých při realizaci aplikace. Dále jsou zde uvedeny některé konkrétní způsoby realizace jednotlivých částí.

Aplikace je implementována jako internetová. Pro správný chod tím pádem uživateli postačuje běžný internetový prohlížeč a spojení se serverem, kde aplikace běží. Veškerá údržba aplikace se obejde bez nutnosti aktualizace na straně uživatelů, protože se provádí pouze na serveru. Použité nástroje a některé konkrétní postupy popsány v následujících kapitolách.

### 7.1 Značkovací jazyk (X)HTML

Značkovací jazyky pro vytváření webových stránek začala specifikací HTML 2.0, který vydala standartizační společnost IETF. Jednalo se o jakousi podmnožinu SGML, což byl univerzální značkovací jazyk, který si můžeme představit jako nástroj k definici jakýchkoli jazyků pro popis informací.

Pro tvorbu webových stránek byl však jazyk HTML 2.0 příliš strohý. Jazyk umožňoval definovat strukturu dokumentu, avšak nějaké barevné nebo grafické ztvárnění stránek neumožňoval. Z tohoto důvodu byl jazyk rozšířen o prvky sloužící k definici vzhledu stránky a vydán jako verze HTML 3.2. Od této verze se vývoje jazyka HTML ujímá konsorcium W3C. Následná verze HTML 4.0 byla soustředěna na maximální možnosti definování struktury dokumentu. Formátování a vzhled dokumentu byl přenechán na jazyku CSS. Později byl vývoj jazyka ukončen vydáním revidované verze HTML 4.01, která se stala výchozím bodem následujících jazyků pro tvorbu webu. Dalším počínáním W3C bylo prosadit jazyk XML jako hlavní značkovací jazyk nejen pro webové dokumenty. Výsledkem byl jazyk XHTML, který specifikací odpovídá revidovanému jazyku HTML 4.01 a integruje pravidla jazyku XML. [8]

Nejnovějším představitelem značkovacích jazyků je verze HTML 5, které zavádí a vylepšuje mnoho funkcí. Obsahuje nástroje pro práci s formuláři, API, multimédií, strukturami a sémantikou. Také má oproti předchozím verzím přesně definováno, jak se

má dokument generovat v případě syntaktické chyby. [9]

Při tvorbě aplikace byla použita dnes již tradiční metoda oddělení struktury a obsahu od vzhledu aplikace, a samozřejmě byl kladen důraz na validní HTML kód dokumentů. Tím by mělo být zaručeno, že běžně používané prohlížeče nebudou mít problémy správně zobrazovat stránky aplikace. Na obrázku 15 je zobrazena tabulka vytvořená pouze značkovacím jazykem bez použití stylování vzhledu.

Název	Vedoucí	Začátek	Konec	Úkoly	Hotovo	
<a href="#">Projekt 01</a>	<a href="#">Karel Kozelka</a>	8.03.2012	23.03.2012	2/15	Ne	<input type="checkbox"/>
<a href="#">test barvy</a>	<a href="#">Karel Kozelka</a>	22.03.2012	22.03.2012	0/0	Ne	<input type="checkbox"/>
<a href="#">A zase je tu neděle</a>	<a href="#">Karel Kozelka</a>	22.03.2012	22.03.2012	0/0	Ne	<input type="checkbox"/>
<a href="#">Jak na krtky</a>	<a href="#">Karel Kozelka</a>	25.03.2012	25.03.2012	0/0	Ne	<input type="checkbox"/>
<a href="#">Test podcastů</a>	<a href="#">Karel Kozelka</a>	5.05.2012	5.05.2012	0/1	Ne	<input type="checkbox"/>
<input type="button" value="Smazat vybrané projekty"/>						

Obrázek 15: Tabulka bez stylování vzhledu

## 7.2 Kaskádové styly CSS

CSS je jazyk popisující způsob zobrazení webových stránek. Kaskádové styly si můžeme představit jako šablony. Při změně šablony se mění vzhled dokumentu, ale obsah a struktura zůstávají zachovány. Byl vyvinut organizací W3C v rámci potřeby oddělit vzhled dokumentu od struktury a obsahu. [10]

Na obrázcích 15 a 16 je dobře viditelné, jaký mají kaskádové styly vliv na stylování dokumentů. Pomocí kaskádových stylů lze s vzhledem dokumentu provádět téměř cokoli. Ať už skrývání částí dokumentů, rozmisťování prvků nebo i různé dynamické efekty reagující například na kurzor myši. Také je možné použít více stylů pro jednu webovou stránku, například zobrazení pro tisk. Používání kaskádových stylů oproti samotnému HTML přináší výhody:

- mnohem rozsáhlejší možnosti formátování
- jednodušší změny vzhledu
- urychlení načítání webových stránek
- možnost existence různých stylů

Hlavní nevýhodou kaskádových stylů je rozdílná interpretace některých vlastností v různých prohlížečích. Některé prohlížeče také nepodporují veškeré vlastnosti CSS verze 2.1. Zejména prohlížeč Microsoft Internet Explorer je specifický svými problémy s odlišným vzhledem nastýlovaných stránek.

Na obrázku 16 je uvedena tabulka výpisu projektů nastýlovaná pomocí kaskádových stylů. V porovnání s obrázkem 15, kde je totožná tabulka bez stylování, je vidět, jak je možné pomocí stylů informace zpřehlednit.

Název	Vedoucí	Začátek	Konec	Úkoly	Hotovo	
Projekt 01	Karel Kozelka	8.03.2012	23.03.2012	2/15	Ne	<input type="checkbox"/>
test barvy	Karel Kozelka	22.03.2012	22.03.2012	0/0	Ne	<input type="checkbox"/>
A zase je tu neděle	Karel Kozelka	22.03.2012	22.03.2012	0/0	Ne	<input type="checkbox"/>
Jak na krtky	Karel Kozelka	25.03.2012	25.03.2012	0/0	Ne	<input type="checkbox"/>
Test podcastů	Karel Kozelka	5.05.2012	5.05.2012	0/1	Ne	<input type="checkbox"/>

Smazat vybrané projekty

Obrázek 16: Tabulka nastýlovaná pomocí CSS

## 7.3 PHP

PHP je velmi populární skriptovací jazyk pro tvorbu dynamických webových aplikací. Je zpracováván na straně serveru a výstupem bývá webová stránka, kterou server vygeneruje a pošle do klienta prohlížeče. Syntaxe jazyku PHP vychází z programovacích jazyků Java, C, Pascal a Perl. Dnes je jazyk PHP velice rozšířenou a oblíbenou technologií hlavně z důvodu své jednoduchosti. [11]

Typický PHP skript obsahuje HTML i PHP kód. Zpracování takového skriptu provádí webový server po obdržení požadavku. Nejprve je proveden PHP kód, poté se k němu připojí nezměněný HTML kód a výsledek odešle jako běžnou HTML stránku, bez jakéhokoli PHP kódu. [12]

Při vývoji aplikace byla využita možnost PHP inkludování souborů. Veškeré funkce generující různé části aplikace byly psány do jednoho PHP souboru. Následné generování konkrétních stránek aplikace probíhá voláním funkcí z inkludovaného PHP souboru. Tento způsob umožňuje jednoduché vytváření nových částí aplikace, které jsou v podstatě pouze složeny z existujících funkcí.

## 7.4 Javascript

Javascript je objektově orientovaný skriptovací jazyk. Jeho kód se vkládá do internetových stránek a je interpretován na straně klienta. Užívá se k vytváření doplňků webových stránek. Například malé programky, vyskakovací okna, kontroly formulářů před odesláním na server, interaktivní galerie a spoustu dalších funkcí.

Pomocí javascriptu jsou však často realizovány prvky, u kterých je to značně nevhodné. Například navigace webové stránky realizovaná javascriptem je nepřístupná vyhledávačům a také uživatelům, jejichž prohlížeč javascript nepodporuje, případně ho mají zakázány. Z tohoto důvodu by se měl javascript používat pouze na doplňkové funkce, které přímo neohroží správnou funkčnost webové stránky.

V aplikaci bylo realizováno pomocí javascriptu několik funkcí, které slouží pro pohodlnější užívání. Pro jednodušší vybírání dat byl implementován javascriptový kalendář clean calendar, distribuovaný pod MIT licenci. Dále byly použity WYSIWYG editor Xinha, dostupný pod BSD licenci, pro více možností textových polí. Tento editor má možnost obsahovat velké množství funkcí. Z důvodu přehlednosti jsou však v aplikaci dostupné pouze základní operace, mezi něž patří různé možnosti formátování písma (velikost, tučné, kurzíva, podtržení, barva, přeškrtnutí), vkládání odkazů, měnitelná barva pozadí a vypnutí editoru.

Pro zobrazování ganttova diagramu byl implementován nástroj jsGantt 1.2, který je přístupný pod BSD licenci. V distribuované verzi však nebyl tento nástroj zcela vhodný pro potřeby aplikace a obsahoval několik chyb, z těchto důvodů muselo být provedeno několik úprav. Nejdůležitější byla úprava správného řazení položek pro správné vykreslování. Dále byla provedena česká lokalizace, odstranění nepotřebných částí a drobné kosmetické úpravy.

## 7.5 Databáze MySQL

MySQL je relační databázový systém, oblíbený zejména při vývoji webových aplikace ve spojení se skriptovacím jazykem PHP. Mezi hlavní výhody patří rychlost, výkonnost, uživatelská jednoduchost a možnost užívání pod Open Source licencí. Tento systém používá standartní dotazovací jazyk SQL. V MySQL je možnost volby z několika způsobů uložení dat. Tabulky typu MyISAM umožňují rychlejší práci s databází, full-textové indexování a vyhledávání [13]. Podporu referenční integrity poskytuje MySQL pouze v případě tabulek typu InnoDB, které naopak neumožňují full-textové vyhledávání [14]. Proto je důležité optimálně zvolit typ uložení dat dle potřeby.

Aplikace využívá tabulek typu InnoDB, které umožňují řešit referenční integritu na úrovni databáze. Správný návrh datového modelu pak zaručuje, že vymazání nebo editace jsou promítnuty do propojených tabulek.

## 7.6 Konkrétní způsoby realizace

V této části práce jsou popsány některé způsoby řešení, které byly při realizaci aplikace použity. Pro lepší představu jsou připojeny i některé části kódu.

### 7.6.1 Přihlašování

Uživatel má přístup do aplikace až po úspěšném přihlášení. Samotná autentizace je řešena obvyklým způsobem, kdy uživatel zadá své přihlašovací údaje (login a heslo) do formuláře. Přihlašovací funkce, které jsou tyto údaje předány, je porovná s údaji v databázi. Pokud zadané přihlašovací údaje odpovídají údajům v databázi, pak jsou vráceny informace o ID uživatele a případných oprávnění administrátora nebo vedoucího. Tyto informace jsou pak uloženy do superglobální proměnné `$_SESSION['uzivatel']`, která je dále předávána každé následující stránce, kde je session nastartovaná pomocí funkce `session_start`. Nakonec je uživatel přesměrován do samotné aplikace.

```

if ($login != "" && $password != "") {
    $password = md5($password);
    db_connect();
    $query = "SELECT uid,vedouci,admin FROM uzivatele WHERE (login LIKE '"
        . mysql_real_escape_string($login) . "')
        AND (heslo LIKE '" . mysql_real_escape_string($password) . "')";
    $result = mysql_query($query);
    $valid = mysql_num_rows($result);
    if ($valid == 1) {
        $prihlas_id = mysql_result($result, 0, "uid");
        $vedouci = mysql_result($result, 0, "vedouci");
        $admin = mysql_result($result, 0, "admin");
        Session_Start();

        $_SESSION['uzivatel'] = Array(
            'userid' => $prihlas_id,
            'vedouci' => $vedouci,
            'admin' => $admin,
        );
        MySQL_Close();
        Header('Location: home_user.php');
    } else {

```

Obrázek 17: Ukázka ověřování přihlašovacích údajů

## 7.6.2 Úprava dat pro Ganttův diagram

Pro zobrazování Ganttova diagramu bylo nutno správně realizovat posloupnost výpisu informací, aby nedocházelo k chybnému zobrazování informací na časové ose. Použitý nástroj Jsgantt byl navržen tak, že ID zobrazovaných úkolů v sobě obsahuje informace, zda se jedná o podúkol nebo nikoliv. Tato konstrukce byla pro aplikaci nevhodná. Bylo tedy nutné ji odstranit a realizovat způsobem, který je pro aplikaci vhodný.

Nejprve jsou z databáze vybrány veškeré úkoly a podúkoly náležící k projektu, ke kterému má být Ganttův diagram vykreslen. U každého úkolu jsou vybrány informace, které budou v diagramu zobrazeny a je zjišťováno, jestli se jedná o podúkol nebo úkol. To je provedeno ověřením existence předka úkolu. Dále je nutno rozlišit, zda vykreslovaný úkol má podúkoly. Pro toto rozlišení je zde posílán sql dotaz databázi, jestli nějaké záznamy s úkoly mají uvedené *id\_predka* shodné s ID úkolu, který má být vykreslen. Dle následné odpovědi je nastaven proměnná *\$group*, která ovlivňuje způsob vykreslení.



```

for ($i = 0; $i < $num_result; $i++) {
    $id_ukolu = mysql_result($result, $i, "id");
    $nazev_ukolu = mysql_result($result, $i, "nazev");
    $zacatek = date("j/m/Y", strtotime(mysql_result($result, $i, "zacatek")));
    $konec = date("j/m/Y", strtotime(mysql_result($result, $i, "konec")));
    $shotovo = mysql_result($result, $i, "shotovo");
    $shotovo = $shotovo * 100;
    $predek = mysql_result($result, $i, "id_predka");
    $group = 0;

    if (is_null($predek)) {
        $predek = 0;
    }
    $query_group = "select * from ukoly where id_predka="
        . mysql_real_escape_string($id_ukolu);
    $result_group = mysql_query($query_group);
    if (mysql_num_rows($result_group) > 0) {
        $group = 1;
    }
}

```

*Obrázek 18: Identifikace hierarchie ganttova diagramu*

### 7.6.3 Soubory na serveru

Soubory odesílané na server ze strany uživatelů jsou ukládány do stromové struktury. Jednotlivé adresáře v této struktuře jsou pojmenovány dle ID projektu nebo úkolu, ke kterému soubor náleží. Adresáře představující úkoly jsou vytvářeny jako podadresáře projektu, ke kterému náleží. V případě podúkolu je adresář vytvořen vnořeně v adresáři úkolu, pod který přísluší. Tímto způsobem jsou soubory na serveru uloženy v daleko přehlednější formě.

Jelikož jsou v databázi uloženy pouze informace o souboru, a ne soubor samotný, je při mazání projektu nebo úkolu nutné mazat i jim přiřazené soubory. Použitím výše popsaného způsobu ukládání souborů pak stačí zavolat funkci pro rekurzivní mazání adresáře se soubory viz obrázek 19.

Funkce nejprve ověří existenci předaného jména adresáře. Pokud adresář existuje, pak je z něj načtena první položka, u které je ověřován typ. Pokud se jedná o soubor, je smazán, a následně je načtena nová položka. V případě adresáře volá funkce sama sebe na mazání nalezeného adresáře. Pokud již neexistuje žádná položka v adresáři, tak je i on samotný smazán.

```

function delete_directory($dirname) {
    if (is_dir($dirname))
        $dir_handle = opendir($dirname);
    if (!$dir_handle)
        return false;
    while ($file = readdir($dir_handle)) {
        if ($file != "." && $file != "..") {
            if (!is_dir($dirname . "/" . $file))
                unlink($dirname . "/" . $file);
            else
                delete_directory($dirname . '/' . $file);
        }
    }
    closedir($dir_handle);
    rmdir($dirname);
    return true;
}

```

Obrázek 19: Funkce pro mazání adresářů

#### 7.6.4 Ověřování vstupních polí formulářů

V aplikaci je použito několik formulářů pro přidávání projektů, úkolů, uživatelů, komentářů a podobně. Některá vstupní pole jsou ošetřena, aby bez nich nebylo možné formulář odeslat.

V aplikaci se také vyskytuje několik vstupních polí pro zadávání dat. Každý uživatel může být zvyklý zapisovat jiný formát data, ale aplikace požaduje jednotný způsob zadávání „dd.mm.rrrr“. Tyto pole jsou automaticky vyplňovány aktuálním datem, díky čemuž uživatel vidí, jaký formát je požadován. Při zapnutém javascriptu je možné vybírat datum z javascriptového kalendáře, který následně pole vyplní datem ve správném formátu. Pro případ vypnutého javascriptu je pole možno vyplnit ručně.

Oba typy vstupních polí jsou ošetřovány na straně serveru pomocí PHP. Bylo by možné ošetření ze strany javascriptu, aby nedocházelo ke zbytečným požadavkům na serveru. Problém by nastal při vypnutém javascriptu, kdy by kontrola neproběhla a bylo by možné odeslat formulář s nesprávnými informacemi.

## 8. Validace

Hotová aplikace byla zaslána firmě TRW Automotive v Jablonci nad Nisou pro otestování a případné připomínky či návrhy dalších rozšíření. Bylo potřeba také získat zpětnou vazbu co se týče kvality aplikace, splnění požadované funkčnosti a zjistit, zda aplikace odpovídá minimálním možným požadavkům, které by na software tohoto typu mohl vznést případný zadavatel. Firma TRW Automotive rovněž pracuje s projektovým řízením, proto byla vhodným kandidátem pro ohodnocení této aplikace.

Aplikaci hodnotili jako velmi dobrý a jednoduchý základ pro řízení softwarových projektů. Navrhli také následující rozšíření aplikace:

- možnost zobrazení Ganttova diagramu, ve kterém by byly uvedeny všechny probíhající projekty,
- Ganttův diagram zobrazující pouze veškeré úkoly a podúkoly,
- možnost přiřazovat projektům a úkolům priority a následné třídění dle priorit,
- emailová upozornění,
- full-textové vyhledávání,
- automatické vyplňování přihlašovacích údajů,
- přidání uživatele s právy pouze pro prohlížení,
- možnost přidávat projekty a úkoly z domovské stránky,
- více základních informací k projektům (zákazník, platforma, ...).

Žádná z těchto navrhovaných rozšíření nebyla implementována do aplikace, a to z důvodu relativně pozdního obdržení posudku od firmy.

Také byla nalezena chyba v aplikaci, kvůli které se neprovádělo přesměrování na požadovanou stránku při editaci nebo vytváření projektu. Byla zapříčiněna vypisováním údajů před příkazem k přesměrování, což nedovolilo tento příkaz provést.

## 9. Závěr

Cílem této bakalářské práce bylo vytvořit pilotní verzi aplikace pro řízení softwarových projektů, která bude obsahovat minimální požadovanou funkcionalitu ale bude zachovávat uživatelsky příjemné prostředí. K tomu bylo potřeba obeznámit se s projektovým řízením, postupy při řízení softwarových projektů a jejich programovou podporou. Následně bylo třeba vytvořit samotnou aplikaci, k čemuž byly použity značkovací jazyk HTML, kaskádové styly CSS, skriptovací jazyky PHP a javascript a databáze MySQL. Bakalářská práce obsahuje rovněž popis jednotlivých částí aplikace doplněný obrázkovými náhledy pro lepší názornost. V závěru práce je připojeno ověření obsahu a funkčnosti aplikace ze strany firmy TRW Automotive.

Z posudku firmy TRW Automotive vyplývá, že se podařilo vytvořit základ pro aplikaci k řízení softwarových projektů, která je schopna pokrýt minimální potřeby. Také byla navržena množina případných dalších rozšíření, které by vedly k zlepšení kvality aplikace a jejímu možnému užití v praxi.

## 10. Seznam literatury

- [1] MANNOVÁ, B. a K. VOSÁTKA. *Řízení softwarových projektů*. Praha: České vysoké učení technické, 2007. ISBN 978-80-01-03297-8.
- [2] KŘENA, B. a KOČÍ, R.: *Úvod do softwarového inženýrství IUS [Studijní opora]*. Dokument dostupný na URL <http://www.scribd.com/doc/48636337/21/Modely-%C5%BEivotního-cyklu-software> (prosinec 2006)
- [3] FIALA, P.: *Řízení projektů*. Praha: Oeconomica, 2008. ISBN 978-80-245-1413-0.
- [4] SVOZILOVÁ, A. *Projektový management*. Praha: Grada, 2006. ISBN 80-247-1501-5.
- [5] *Metodika projektového řízení objektově orientovaného vývoje*. Dokument dostupný na URL <http://www.fit.vutbr.cz/study/courses/MPR/public/metodika-oo/index.html> (květen 2012)
- [6] ROSENAU, M. D. *Řízení projektů : příprava a plánování, zahájení, výběr lidí a jejich řízení, kontrola a změny, vyhodnocení a ukončení*. Brno: Computer Press, 2003. ISBN 80-7226-218-1.
- [7] Schwalbe K.: *Řízení projektů v IT*, Brno: Computer Press, 2007, ISBN 978-80-251-1526-8.
- [8] XHTML-vývoj (X)HTML a jeho možnosti. *Interval.cz* [online]. 2002 [cit. 2012-05-17]. Dostupné z: <http://interval.cz/clanky/xhtml-vyvoj-xhtml-a-jeho-moznosti/>
- [9] XHTML je mrtvé! Ať žije HTML5! Nebo ne?. *Zdroják.cz* [online]. 2011 [cit. 2012-05-17]. Dostupné z: <http://www.zdrojak.cz/clanky/xhtml-je-mrtve-at-zije-html5-nebo-ne/>
- [10] Kaskádové styly. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-05-17]. Dostupné z: [http://cs.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://cs.wikipedia.org/wiki/Cascading_Style_Sheets)
- [11] PHP-Historie a budoucnost. *Linuxsoft.cz* [online]. 2004 [cit. 2012-05-17]. Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=171](http://www.linuxsoft.cz/article.php?id_article=171)

- [12] PHP-Jak to funguje. *Linuxsoft.cz* [online]. 2004 [cit. 2012-05-17]. Dostupné z: [http://www.linuxsoft.cz/article.php?id\\_article=172](http://www.linuxsoft.cz/article.php?id_article=172)
- [13] The MyISAM Storage Engine. *MySQL* [online]. 2012 [cit. 2012-05-17]. Dostupné z: <http://dev.mysql.com/doc/refman/5.5/en/myisam-storage-engine.html>
- [14] The InnoDB Storage Engine. *MySQL* [online]. 2012 [cit. 2012-05-17]. Dostupné z: <http://dev.mysql.com/doc/refman/5.5/en/innodb-storage-engine.html>